back-end – Estrutura e Inicialização do Servidor

O back-end do sistema CineMatch foi desenvolvido utilizando Node.js com Express, sendo responsável por disponibilizar uma API REST para comunicação com o front-end.

Arquivo server.js

O arquivo server.js é o ponto de entrada da aplicação, responsável por inicializar o servidor e configurar os principais middlewares do sistema.

As principais responsabilidades desse arquivo são:

Carregar variáveis de ambiente através do pacote dotenv

Inicializar o servidor Express

Configurar middlewares globais

Registrar as rotas da aplicação

Iniciar o servidor na porta configurada para o ambiente de execução

Os middlewares globais utilizados são:

cors: permite comunicação entre diferentes origens

express.json(): permite o envio e recebimento de dados no formato JSON

O servidor expõe uma rota raiz (/) utilizada apenas para verificação do funcionamento da API.

As rotas do sistema são organizadas por domínio e registradas da seguinte forma:

/auth – autenticação de usuários

/users – operações relacionadas ao usuário

/filmes – gerenciamento e consulta de filmes

/listas – gerenciamento de listas personalizadas

O servidor é inicializado utilizando a porta definida na variável de ambiente PORT, com valor padrão configurado para ambiente local.

back-end – Integração com Banco de Dados

Arquivo supabase.js

O sistema utiliza o Supabase, uma plataforma baseada em PostgreSQL, como banco de dados principal do projeto.

O arquivo supabase.js é responsável por centralizar a criação do cliente de conexão com o banco de dados, utilizando as variáveis de ambiente:

SUPABASE_URL

SUPABASE_SERVICE_ROLE_KEY

Essa abordagem permite que o cliente Supabase seja reutilizado em diferentes partes da aplicação, como controllers e serviços, garantindo:

Organização do código

Facilidade de manutenção

Separação entre lógica de negócio e acesso a dados

back-end – Controller de Autenticação

O arquivo auth.controller.js é responsável por concentrar as funcionalidades de autenticação de usuários do sistema CineMatch.

Esse controller gerencia os processos de cadastro, login e validação de acesso a rotas protegidas, utilizando autenticação baseada em JWT.

Cadastro de Usuário (register)

A funcionalidade de cadastro permite que novos usuários criem uma conta no sistema.

Durante esse processo, o sistema executa as seguintes etapas:

Validação dos dados obrigatórios (nome, e-mail e senha)

Normalização do e-mail para evitar duplicidade

Criptografia da senha utilizando o algoritmo bcrypt

Persistência dos dados do usuário no banco de dados via Supabase

Após o cadastro, o sistema retorna apenas informações não sensíveis do usuário, garantindo a segurança dos dados.

Login de Usuário (login)

O processo de login é responsável por autenticar usuários já cadastrados.

As etapas executadas são:

Normalização do e-mail informado

Busca do usuário no banco de dados

Validação da senha informada com o hash armazenado

Geração de um token JWT com tempo de expiração definido

O token gerado é utilizado para autenticar requisições futuras realizadas pelo usuário.

Middleware de Autenticação (authMiddleware)

Além dos controllers de login e cadastro, o módulo de autenticação também disponibiliza um middleware de validação JWT, responsável por proteger rotas sensíveis do sistema.

Esse middleware:

Verifica a presença do token no header Authorization

Valida o token utilizando a chave secreta do ambiente

Decodifica os dados do usuário autenticado

Autoriza ou bloqueia o acesso à rota solicitada

Esse mecanismo garante que apenas usuários autenticados tenham acesso a funcionalidades restritas do sistema.

Ajustes em Relação ao Planejamento Inicial

Durante o desenvolvimento do back-end, optou-se por centralizar os controllers de autenticação e o middleware JWT em um único módulo, simplificando a organização do código e facilitando a manutenção do sistema.

back-end – Rotas de Autenticação

Arquivo auth.routes.js

O arquivo auth.routes.js define as rotas públicas responsáveis pela autenticação de usuários no sistema CineMatch.

As rotas disponíveis são:

POST /auth/register

Responsável pelo cadastro de novos usuários.

Recebe os dados básicos do usuário e delega o processamento ao controller de autenticação.

POST /auth/login

Responsável pela autenticação de usuários já cadastrados.

Retorna um token JWT utilizado para acesso às rotas protegidas do sistema.

Essas rotas não utilizam middleware de autenticação, pois são o ponto inicial de acesso ao sistema.

back-end – Rotas de Usuário

Arquivo users.routes.js

O arquivo users.routes.js concentra as funcionalidades relacionadas ao perfil do usuário autenticado.

Todas as rotas deste módulo são protegidas pelo middleware de autenticação JWT.

Rotas implementadas:

GET /users/me

Retorna os dados do usuário autenticado, como nome, e-mail e foto de perfil.

PUT /users/me

Permite a atualização dos dados do usuário, incluindo nome, e-mail, senha e foto.

Caso a senha seja alterada, ela é novamente criptografada antes de ser salva.

POST /users/me/avatar

Permite o upload da foto de perfil do usuário.

O arquivo é armazenado no Supabase Storage, e a URL pública é salva no banco de dados.

DELETE /users/me/avatar

Remove a foto de perfil do usuário tanto do storage quanto do banco de dados.

Esse módulo garante o gerenciamento completo do perfil do usuário dentro do sistema.

back-end – Rotas de Listas Personalizadas

Arquivo listas.routes.js

O arquivo listas.routes.js é responsável pelo gerenciamento das listas personalizadas de filmes criadas pelos usuários.

Todas as rotas são protegidas pelo middleware de autenticação.

Funcionalidades disponíveis:

POST /listas

Cria uma nova lista vinculada ao usuário autenticado.

GET /listas

Lista todas as listas pertencentes ao usuário, ordenadas por data de criação.

POST /listas/:listaId/filmes

Adiciona um filme a uma lista específica.

GET /listas/:listaId/filmes

Retorna todos os filmes associados a uma lista, utilizando relacionamento entre tabelas no banco de dados.

DELETE /listas/:id

Remove uma lista pertencente ao usuário autenticado.

DELETE /listas/:listaId/filmes/:filmeId

Remove um filme específico de uma lista.

Esse conjunto de rotas possibilita a criação, organização e manutenção de listas personalizadas no sistema.

back-end – Rotas de Filmes

Arquivo filmes.routes.js

O arquivo filmes.routes.js gerencia o armazenamento local dos filmes salvos no sistema, evitando duplicidade de dados vindos da API externa (TMDB).

Todas as rotas utilizam autenticação JWT.

Rotas implementadas:

POST /filmes

Cria um registro de filme no banco de dados caso ele ainda não exista. Antes da inserção, o sistema verifica se o filme já foi salvo anteriormente.

GET /filmes

Retorna todos os filmes salvos no banco de dados, ordenados por data de criação.

DELETE /filmes/:id

Remove um filme do banco de dados.

Esse módulo permite centralizar os filmes utilizados nas listas e nas funcionalidades do sistema.

Limitações e Decisões de Projeto

Durante o desenvolvimento do projeto CineMatch, algumas decisões técnicas foram tomadas com o objetivo de garantir a entrega de um sistema funcional.

Embora o planejamento previsse uma arquitetura completa de autenticação baseada em JWT com refresh tokens, optou-se por implementar apenas o JWT tradicional, sem o uso de refresh tokens. Essa decisão foi motivada pela complexidade adicional de gerenciamento de múltiplos tokens, o que extrapolaria o escopo necessário para a proposta do projeto, sem comprometer a segurança básica da aplicação.

A autenticação de usuários foi implementada utilizando JWT e criptografia de senhas com bcrypt, atendendo de forma adequada aos requisitos de segurança esperados para um sistema acadêmico, garantindo confidencialidade das credenciais e controle de acesso às rotas protegidas.

Em relação à infraestrutura, o back-end foi hospedado utilizando a plataforma Render, escolhida por sua facilidade de deploy, integração nativa com aplicações Node.js e suporte a variáveis de ambiente, permitindo a publicação do sistema de forma simples e eficiente para fins de validação e demonstração do projeto.

Algumas funcionalidades inicialmente consideradas, como mecanismos mais avançados de escalabilidade e controle de sessão, não foram implementadas, uma vez que o foco principal do projeto está na organização das funcionalidades essenciais do sistema, como autenticação, gerenciamento de usuários, listas personalizadas e integração com serviços externos.

Dessa forma, as decisões adotadas priorizaram a simplicidade, a clareza arquitetural e a aderência aos objetivos educacionais do projeto CineMatch.

Da mesma forma, o back-end foi projetado exclusivamente para o gerenciamento de filmes, não contemplando séries ou outros tipos de mídia. Essa escolha permitiu uma modelagem de dados mais simples, rotas mais objetivas e melhor integração com a API externa TMDB, atendendo plenamente aos objetivos do projeto sem comprometer sua estabilidade e clareza arquitetural

Front-end Web – Implementação Final

O front-end web do sistema CineMatch foi desenvolvido utilizando HTML, CSS e JavaScript puro, adotando uma arquitetura baseada no consumo de uma API REST, responsável por se comunicar diretamente com o back-end da aplicação.

Embora no planejamento inicial estivesse prevista a utilização de frameworks modernos de front-end, durante o desenvolvimento optou-se por uma abordagem mais direta, priorizando simplicidade, controle total do fluxo da aplicação e melhor compreensão do funcionamento interno do sistema.

Essa decisão permitiu:

Maior controle sobre o comportamento da aplicação sem dependência de bibliotecas externas

Redução da complexidade estrutural do projeto

Facilidade na integração com a API REST desenvolvida em Node.js

Melhor alinhamento com os objetivos acadêmicos do trabalho, com foco nos fundamentos do desenvolvimento web

Estrutura do Front-end

O front-end do CineMatch é organizado em arquivos independentes, cada um com responsabilidade bem definida:

Arquivos HTML: responsáveis pela estrutura e organização das páginas

CSS próprio (style.css): responsável pela estilização visual e responsividade da interface

JavaScript (ES6+): responsável pela lógica da aplicação, autenticação, consumo de APIs e manipulação do DOM

Essa separação garante melhor organização, legibilidade do código e

facilidade de manutenção.

Autenticação no Front-end

A autenticação do usuário é realizada por meio de JWT (JSON Web Token), recebido após o login e armazenado no localStorage do navegador.

O fluxo de autenticação contempla:

Cadastro de novos usuários

Login com validação de credenciais

Persistência da sessão do usuário

Proteção de páginas autenticadas por meio de verificação do token

Caso o token não exista, o usuário é automaticamente redirecionado para a página de login.

Consumo de APIs

O front-end consome duas fontes principais de dados:

API própria do CineMatch (Node.js + Express)

Responsável pelo gerenciamento de usuários, listas personalizadas, filmes salvos e favoritos.

API externa TMDB (The Movie Database)

Utilizada para obtenção de informações de filmes, como títulos, sinopses, gêneros, diretores, elenco e imagens.

Os dados obtidos são processados e exibidos dinamicamente, proporcionando uma experiência interativa ao usuário.

Gerenciamento de Estado

Mesmo sem o uso de bibliotecas de estado global, o sistema realiza o controle de estado de forma eficiente por meio de:

Variáveis em memória durante a execução das páginas

Cache local de filmes favoritos

localStorage para persistência do token de autenticação

Essa abordagem mostrou-se suficiente para o escopo do projeto, mantendo clareza, previsibilidade e simplicidade no fluxo de dados.

Estilização e Interface

A interface do usuário foi desenvolvida com CSS próprio, utilizando conceitos de responsividade, hierarquia visual e usabilidade.
O layout foi planejado para oferecer uma experiência clara e intuitiva, alinhada à proposta visual do sistema CineMatch, com foco em:
Facilidade de navegação
Organização das informações
Aparência moderna e consistente

Ajustes em Relação ao Planejamento Inicial

Durante o planejamento inicial do projeto, previa-se a utilização de frameworks modernos de front-end. No entanto, após análise do escopo real da aplicação e das necessidades do sistema, optou-se por uma implementação utilizando apenas tecnologias nativas da web.
Essa escolha permitiu maior transparência no funcionamento da aplicação, facilitando o entendimento do fluxo completo entre front-end e back-end, sem comprometer os requisitos funcionais definidos inicialmente.
Os requisitos funcionais e não funcionais do sistema foram atendidos de forma implícita ao longo da implementação descrita, não sendo listados formalmente, mas contemplados pelas funcionalidades desenvolvidas.

Front-end Web – Páginas Autenticadas e Funcionalidades

Após a autenticação do usuário, o sistema CineMatch disponibiliza um conjunto de páginas responsáveis pela navegação principal, gerenciamento de listas, visualização de detalhes dos filmes e configuração do perfil do usuário. Todas essas páginas se comunicam diretamente com a API REST do back-end e utilizam autenticação baseada em JWT.

Página Inicial Autenticada (Home)

Arquivos envolvidos:
home.html e home.js
A página inicial autenticada é responsável por exibir recomendações de filmes ao usuário logado.
Principais funcionalidades:
Verificação da autenticação por meio do token armazenado no localStorage

Exibição do nome e da foto do usuário no topo da página

Consumo da API do TMDB para obtenção de filmes populares

Navegação para a página de detalhes ao clicar em um filme

Essa página representa o principal ponto de interação do usuário com o sistema.

Página de Detalhes do Filme

Arquivos envolvidos:

detalhes.html e detalhes.js

A página de detalhes apresenta informações completas sobre um filme selecionado.

Funcionalidades implementadas:

Consulta à API do TMDB para obtenção de dados detalhados do filme

Exibição de título, sinopse, gêneros e diretor

Controle de favoritos por meio de um botão interativo

Integração com o back-end para adicionar ou remover filmes da lista

“Favoritos”

Redirecionamento inteligente para a página de origem (Home ou Listas)

Essa página integra dados externos com informações persistidas no sistema.

Página de Listas Personalizadas

Arquivos envolvidos:

listas.html e listas.js

A página de listas permite ao usuário visualizar todas as listas associadas à sua conta.

Funcionalidades:

Listagem das listas vinculadas ao usuário autenticado

Exibição dos filmes pertencentes a cada lista

Navegação direta para a página de detalhes de um filme

Comunicação direta com o back-end para recuperação dos dados

Essa funcionalidade possibilita a organização personalizada dos filmes favoritos.

Página de Configurações da Conta

Arquivos envolvidos:

configuracoes.html e configuracoes.js

A página de configurações permite o gerenciamento completo do perfil do usuário.

Funcionalidades disponíveis:

Visualização dos dados do usuário

Alteração de nome, e-mail e senha

Upload e remoção da foto de perfil

Armazenamento da imagem no Supabase Storage

Encerramento da sessão do usuário (logout)

Todas as alterações realizadas são imediatamente persistidas no back-end.

Integração Geral do Front-end

O front-end do CineMatch foi desenvolvido com foco em:

Separação clara entre estrutura, estilo e comportamento

Comunicação eficiente com APIs REST

Uso consciente de tecnologias nativas da web

Organização modular dos arquivos

Experiência de uso fluida e intuitiva

Mesmo sem a utilização de frameworks como React, a aplicação atende plenamente aos requisitos funcionais propostos, demonstrando domínio dos conceitos fundamentais do desenvolvimento web moderno.

Decisão de Escopo: Foco Exclusivo em Filmes e Recomendações

Durante o planejamento inicial do CineMatch, considerou-se a inclusão de séries, filtros avançados e barra de pesquisa. No entanto, ao longo do desenvolvimento, optou-se por restringir o escopo da aplicação exclusivamente à recomendação de filmes, removendo funcionalidades como busca manual e filtros complexos.

Essa decisão foi tomada porque o objetivo central do sistema é atuar como uma plataforma de recomendação, onde o usuário não precisa procurar ativamente um conteúdo, mas sim receber sugestões automáticas com base em

popularidade e interações anteriores.

A exclusão da busca e dos filtros reforça a proposta de uso do sistema, incentivando a descoberta de novos filmes de forma intuitiva, reduzindo a complexidade da interface e evitando que o usuário dependa de pesquisas manuais, como em catálogos tradicionais.

Quanto às séries, optou-se por não incluí-las para manter consistência de dados, simplicidade na modelagem do back-end e foco no escopo principal do projeto, uma vez que filmes e séries possuem estruturas distintas de duração, temporadas e episódios, o que aumentaria significativamente a complexidade do sistema.

Dessa forma, a ausência de busca manual e filtros avançados está diretamente relacionada à proposta original de recomendação automática, que seria potencializada com a futura implementação de um algoritmo mais robusto.

Front-end Mobile – Progressive Web App (PWA)

O front-end mobile do sistema CineMatch foi implementado no formato de Progressive Web App (PWA), utilizando HTML, CSS e JavaScript puro, compartilhando a mesma API REST do back-end da aplicação.

Embora no planejamento inicial estivesse prevista a utilização de React com Workbox, durante o desenvolvimento optou-se por uma abordagem baseada em tecnologias nativas da web. Essa decisão foi tomada visando redução da complexidade, maior controle da aplicação e melhor adequação ao escopo acadêmico do projeto, sem comprometer os requisitos funcionais propostos.

Essa escolha permitiu a criação de uma aplicação mobile leve, instalável e totalmente integrada ao back-end do sistema.

Arquitetura da PWA

A aplicação mobile é composta pelos seguintes elementos principais:

Service Worker (sw.js)

Web App Manifest (manifest.json)

Arquivos próprios de estilo e lógica mobile (mobile.css e mobile.js)

Essa estrutura possibilita que o sistema funcione como um aplicativo

instalável, acessível diretamente pela tela inicial do dispositivo móvel.

Service Worker

O Service Worker foi implementado no arquivo sw.js e é responsável pelo controle de cache da aplicação.

Funcionalidades implementadas:

Cache dos principais arquivos HTML, CSS e JavaScript

Disponibilização de funcionamento offline limitado

Interceptação de requisições via evento fetch

A estratégia adotada foi Cache First, na qual os recursos são buscados primeiramente no cache e, caso não estejam disponíveis, são recuperados da rede. Essa abordagem garante melhor desempenho e acesso básico à aplicação mesmo sem conexão ativa com a internet.

Web App Manifest

O arquivo manifest.json define as configurações necessárias para que o sistema seja reconhecido como um aplicativo instalável.

Principais configurações:

Nome e nome curto da aplicação (CineMatch)

Inicialização em modo standalone

Definição de cores de tema e fundo

Orientação fixa em modo retrato

Ícone para instalação na tela inicial

Essas configurações permitem que o CineMatch se comporte como um aplicativo mobile quando instalado no dispositivo do usuário.

Autenticação no Mobile

A autenticação no ambiente mobile segue o mesmo padrão do front-end web, utilizando JWT (JSON Web Token).

O fluxo de autenticação contempla:

Cadastro de usuários via API REST

Login com validação de credenciais

Armazenamento do token JWT no localStorage

Envio automático do token no header Authorization

Proteção de páginas autenticadas

Esse modelo garante consistência e segurança entre as versões web e mobile da aplicação.

Página Inicial Mobile (Home)

Arquivos envolvidos:

home.html e home.js

A página inicial do mobile concentra as principais funcionalidades do usuário autenticado.

Funcionalidades implementadas:

Validação do token JWT

Exibição da foto de perfil do usuário

Sistema de abas (Recomendações e Minhas Listas)

Consumo da API do TMDB para exibição de recomendações

Exclusão automática de filmes já favoritados

Favoritar filmes diretamente pela interface mobile

O sistema garante a exibição contínua de recomendações, mantendo sempre um número mínimo de filmes disponíveis ao usuário.

Gerenciamento de Favoritos e Listas

A versão mobile integra-se totalmente com o back-end para gerenciamento de listas personalizadas.

Funcionalidades disponíveis:

Criação automática da lista padrão “Favoritos”

Associação e remoção de filmes das listas do usuário

Visualização das listas com dados complementares da API TMDB

Essa integração garante persistência e sincronização entre as versões web e mobile do sistema.

Página de Configurações Mobile

Arquivos envolvidos:

configuracoes.html e configuracoes.js

Funcionalidades disponíveis:

Visualização de nome e e-mail

Alteração de nome, e-mail e senha

Upload e remoção da foto de perfil

Armazenamento da imagem no Supabase Storage

Encerramento da sessão (logout)

Todas as alterações realizadas são imediatamente persistidas no back-end.

Estilização Mobile

A estilização da versão mobile foi desenvolvida no arquivo mobile.css, com foco em:

Layout vertical responsivo

Navegação por abas

Barra inferior fixa

Hierarquia visual clara

Interface otimizada para toque

O CSS foi desenvolvido especificamente para dispositivos móveis, garantindo melhor usabilidade em telas menores.

Ajustes em Relação ao Planejamento Inicial

Durante o planejamento inicial do projeto, previa-se a utilização de React com Workbox para o desenvolvimento do front-end mobile. Entretanto, ao longo da implementação, optou-se por uma abordagem baseada em tecnologias nativas da web, mantendo o projeto como um PWA leve e funcional.

Essa decisão foi motivada por:

Redução da complexidade do projeto

Maior clareza didática

Facilidade de manutenção

Adequação ao tempo e escopo do projeto acadêmico

Apesar da mudança tecnológica, a versão mobile atende plenamente aos objetivos propostos, oferecendo experiência mobile, instalação no dispositivo e funcionamento offline limitado, totalmente integrada ao back-end do sistema CineMatch.

Algoritmo de Recomendação

Durante o planejamento inicial do CineMatch, previa-se a implementação de um algoritmo de recomendação mais elaborado, baseado em preferências do usuário, histórico de interações, gêneros e diretores, com o objetivo de personalizar de forma mais precisa as sugestões exibidas na plataforma.

No entanto, devido às limitações de tempo e à complexidade envolvida na análise de dados e na modelagem desse tipo de algoritmo, optou-se por não implementar essa funcionalidade em sua forma completa.

Na versão final do sistema, as recomendações são baseadas principalmente em dados fornecidos pela API externa TMDB, como popularidade e relevância, combinados com a exclusão de filmes já favoritados pelo usuário, garantindo uma experiência funcional e coerente com a proposta do sistema.

A implementação de um algoritmo de recomendação mais avançado fica registrada como possível evolução futura do projeto.